JavaScript and jQuery Course
Instructor Teresa Pelkie
Session 1

Topic: The JavaScript Syntax, Hello World

Chapter 1: Introduction to web development – short review

Chapter 2: Getting started with JavaScript up to page 61

Chapter 3: page 93 – using `document.getElementById()` and `document.write()` methods

Overview of JavaScript

What is JavaScript?

- scripting language - runs in the browser, interpreted by the browser, is seen in source
- cross-platform
- **object-oriented**
    - objects - what we work with or manipulate
        - properties - characteristics of an object
        - methods - things an object can do
        - events - things that happen to an object
- weakly typed - the data type is determined by its use
- implemented / standardized by ECMA (European Computer Manufacturers Association), not the W3C
- also used to make interactive PDF documents
- **case sensitive**

JavaScript is *not*

- Java or a subset of Java, although they share common syntax
- about creating special effects, but for improving the user experience

History of JavaScript

- developed by Netscape Corporation in 1996, first called "LiveScript"
- poorly implemented in the early browsers, therefore web developers would use "browser detection" to customize their code for different browsers
- part of the AJAX (Asynchronous JavaScript and XML) technology, which allows the browser to make page requests and changes without refreshing the screen
- used to create many of today's frameworks - a framework is a library of code that saves development time
    - jQuery
    - Dojo
    - Prototype

JavaScript Programming Approaches

Progressive Enhancement

- a website is usable by everyone and by as many devices as possible, with and without JavaScript
- create basic functionality, then add enhancements based on browser capabilities

Unobtrusive JavaScript

- refers to the separation of JavaScript code from the content (HTML) of the web page

Graceful Degradation - not favored today

- the opposite of progressive enhancement
- a fully-featured website is developed first, then considerations are made for browser differences

Three layers of the web page

1. Structure - HTML
2. Presentation - CSS
3. Behavior - JavaScript

Getting Started with JavaScript

Overview of Objects

Objects are the "things" that we manipulate to make interactive web pages.

Browser Object Model or BOM

The *Browser Object Model* (BOM) is the traditional JavaScript object model which contains the built-in objects that JavaScript uses. The names are **case sensitive**. Some of the commonly used objects are:

- window
- document
- Math
- Date

These objects may have *properties* and *methods*. The name of the property or method is appended to the object's name with a period character. This is called the "**dot syntax**".

Properties are *characteristics*. Methods *do something* and are followed by parentheses.

`window.alert()` - page 60, 61

`document.write()` - page 53

The Document Object Model - page 98

The *Document Object Model* (DOM) is a standard set forth by the W3C that describes a way to represent and navigate the web page. It is technically not a part of JavaScript, but JavaScript uses the DOM to interact with HTML elements.

The latest and most trusted version is DOM level 3, which was released in 2004.

If the name of a property or method contains several words, the first letter of each word, excluding the first word, is in uppercase. This is called "**camel-case**".

Example:

```
document.getElementById()
```

JavaScript statements end in a semicolon

```
alert("hello world");
```

JavaScript comments page 58, 59

```
// single line


/*   multiple line
     The comment can continue
     over multiple lines
*/
```

Writing JavaScript Code - using correct HTML5

Make sure that you are using correct HTML5 syntax and include the `doctype` statement (see page 13).

```
<!doctype html>
<html>
<head>
    <meta charset="utf-8">          <!-- goes first -->
    <title>The Page Title</title>   <!-- goes second -->
</head>

<body>
</body>
</html>
```

JavaScript can be added to the HTML document in either the `<head>` or `<body>` sections using the **embedded `<script></script>` tags**

Note: placing JavaScript in the `<body>` is not the preferred approach

Example of placing JavaScript code in an HTML document

```
<!doctype html>
<head>
    <meta charset="utf-8">
    <title>The Page Title</title>

    <script>
       // JavaScript Goes Here - embedded JavaScript
    </script>
</head>

<body>
</body>
</html>
```

JavaScript can be added to the HTML document using an **external file.**

Example:

```
<!doctype html>
<head>
    <meta charset="utf-8">
    <title>The Page Title</title>

    <script src="file.js"></script>  <!-- external file for JavaScript -->
</head>

<body>
</body>
</html>
```

If there is no JavaScript support - see page 52-53

```
<!doctype html>
<head>
    <meta charset="utf-8">
    <title>The Page Title</title>

    <script>
       // JavaScript Goes Here - embedded JavaScript
    </script>
</head>

<body>
    <noscript>Your browser does not support JavaScript</noscript>  <!-- goes in the <body> -->
</body>
</html>
```

Tools for Writing JavaScript

- a plain text editor
- an HTML editor such as **Komodo Edit** or **BlueFish**
- an Integrated Development Environment (IDE) such as Aptana Studio (page 34)
    - generally have better syntax highlighting, debugging and code intelligence
- Firebug plugin for FireFox (page 126)
- online tools
    - www.jsbin.com
    - www.jsfiddle.net

Tools - page 83

- www.jslint.com
- www.jshint.com
- Firebug Add-on
- Editor or IDE such as Aptana Studio

Debugging JavaScript

Common causes of errors - page 120

- Syntax error
    - JavaScript is case sensitive
    - quotations not balanced
    - opening/closing { } and ( ) not balanced
    - using reserved words
- Runtime error
    - reference to something that does not exist
- Logical error
    - error in thinking or expectations

Videos - Please watch

Introduction from lynda.com - nice overview of the language!

http://www.lynda.com/home/Player.aspx?lpk4=87517&playChapter=False

Sample Files for this session

File Name	Description
01_alert.html	Shows how to use the JavaScript `window.alert` method in a `<script>` block
01_alert_from_body.html	Shows how to use the JavaScript `window.alert` method in an `onclick` event
01_document_write.html	Shows how to use `document.write` in a `<script>` block that is used in the `<body>` section of the page
01_dom_anon_function.html	Shows how to use an anonymous function in the `window.onload` event and how to use the DOM to write to an element on the page
01_dom_named_function.html	Shows how to use a named function in the `window.onload` event and how to use the DOM to write to an element on the page