Overview

• What we mean by scripting
• Overview of JavaScript
• Simple examples from http://www.w3schools.com/jsref/dom_obj_event.asp
• Slides are based on textbook: Javascript by Don Gosselin
JavaScript and Client-Side Scripting

• When HTML was first developed, Web pages were static
  – Static Web pages cannot change after the browser renders them

• HTML and XHTML could only be used to produce static documents

• JavaScript
  – Client-side (i.e. executes by the browser) scripting language that allows Web page authors to develop interactive Web pages and sites
The need for scripting

• Static pages are sometime not enough.
  – Content can be dynamic
    • Changing over time or parameter needing setting
    • Timetables, online shops, search engine, library catalogue
    • Form checking

• Client side scripting languages are ones that the browser interprets
JavaScript

• As with other scripting languages, it is *interpreted* instead of compiled
  – Requires more processing
• Weakly typed – you don’t need to declare the *type* of the variable
• Object oriented as well as functional
• Has nothing to do with Java
• Follows the ECMAScript standard
• Is very powerful! (term *scripting language* is misleading)
The `<script>` Element

• Scripts
  – JavaScript programs contained within a web page
  – May be in `<body>` or `<head>`
  – May be in external file
  – Terminated with `</script>`
The `<script>` element...

- Tells the web browser that the scripting engine must interpret the commands it contains
- The `type` attribute tells the browser which scripting language and which version of the scripting language is being used:

  `<script type="text/javascript">`
Placing JavaScript in the Document Head or Document Body

• You can place `<script>` elements in either the document head or document body
• Good idea to place as much of your JavaScript code as possible in the document head
Including a `<script>` Element for Each Code Section

• Include as many script sections as you like within a document

• When you include multiple script sections in a document, you must include a `<script>` element for each section
Creating a JavaScript Source File

• JavaScript source file
  – Usually designated by the file extension .js
  – Does not contain a `<script>` element
  – Cannot include XHTML elements

• To access JavaScript code saved in an external file, assign to the `src` attribute of the `<script>` element the URL of the JavaScript source file

• Use a combination of embedded JavaScript code and JavaScript source files in your documents
Understanding JavaScript Objects

• **Object**
  – Programming code and data that can be treated as an individual unit or component

• **Functions**
  – Individual statements used in a computer program grouped into logical units
  – Used to perform specific tasks

• **Methods**
  – Functions associated with an object
  – For example: `loan.calcPayments();`
Understanding JavaScript Objects (continued)

• **Property**
  – Piece of data associated with an object
  – Assign a value to a property using an equal sign
    
    ```javascript
    loan.interest = .08;
    ```

• **Argument**
  – Information that must be provided to a method
  – Providing an argument for a method is called passing arguments
    
    ```javascript
    loan.calcPayments(800);
    ```
Using the `write()` and `writeln()` Methods

- `document` object represents the content of a browser’s window

- You create new text on a Web page with the `write()` method or the `writeln()` method of the `document` object
  - Both methods require a text string as an argument
    - **Text string** or **literal string**: text that is contained within double or single quotation marks

```javascript
document.write("Bienvenue au Canada!");
```
Simple Example

<body>
  <script type="text/javascript">
    document.write("<p>This is my first JavaScript!</p>" newX);
  </script>
</body>

• Note the ; at the end of the code, something people forget when first stating and will often trip us up.
• The argument of the write method contains html!
Example Explained

• There is an object called the Document
  – The html page

• We can do thing to the document
  – Write to it as in the example
  – These built in action are called methods

• We can find out things about the documents
  – These are its properties.
  – Like its title, URL, date last modified, its cookies.
Case Sensitivity in JavaScript

• JavaScript is case sensitive
• Within JavaScript code, object names must always be all lowercase
Adding Comments to a JavaScript Program

• **Comments**
  – Nonprinting lines that you place in your code to contain various types of remarks

• **Line comment**
  – Hides a single line of code
  – Add two slashes `//` before the comment text

• **Block comments**
  – Hide multiple lines of code
  – Add `/*` before the first character you want included in the block and `*/` after the last character in the block
JavaScript and HTML validation

• Validators do not recognise JavaScript and will therefore give error

• To avoid this, you have to tell the validator that to not parse it, as follows:

```html
<script type="text/javascript">
![CDATA[
...
]]>
</script>
```

• Older browsers do not recognize this, and so you need to place JavaScript comments around it:

```html
/* <![CDATA[ */
/* ]]> */
```
Example Revisited

<body>
<script type="text/javascript">
/* <![CDATA[ */

document.write("<p>This is my first JavaScript!</p>");

/* ]]> */
</script>
</body>
Summary

• Scripting helps make a static page dynamic
• JavaScript runs on the client (browser)