CSCU9A2: Programming and User Interface Design

Web Design 2: Scripting and the DOM

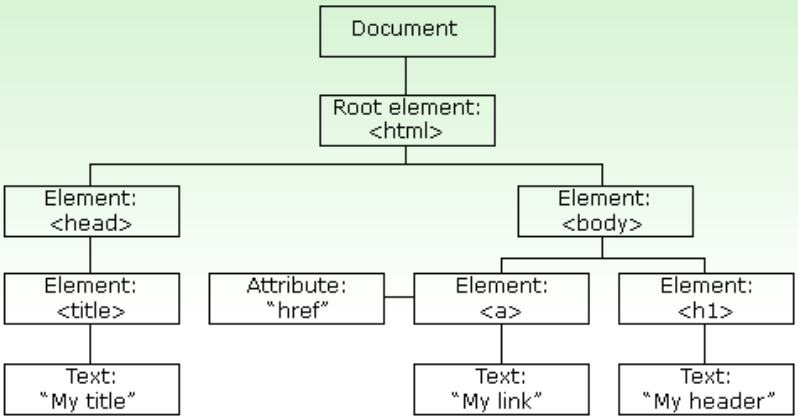
Scripting languages and JavaScript

- A scripting language is a special purpose programming language that automates the execution of tasks associated with a particular software environment.
  - Examples: Job Control Language, shell script, Perl, and many others
  - Programs are typically small and are called scripts (or macros)
- JavaScript is a scripting language that runs within web browsers. Its purpose is to extend the capabilities of HTML.
  - HTML on its own is static: web pages look the same every time they are viewed
  - JavaScript is used to make pages dynamic and interactive
  - A survey of the 1 million most popular websites showed that 92.3% used JavaScript, and its use is growing (w3techs.com, Jan 2013).

How does JavaScript work?

- JavaScript is simply downloaded from the server along with the HTML source
  - and can often be viewed along with the rest of the source of the page
- The JavaScript text is interpreted by the browser (client)
  - client-side processing
- JavaScript sees the whole web page as an object tree
  - As a result it can alter material on the page when it is executed.

HTML Document Object Model (DOM) tree



From: http://www.w3schools.com/htmldom/default.asp

Using JavaScript

- The JavaScript text is incorporated using the `<script></script>`tag
  - `<script>` is like any other HTML tag
- Format:

```
<script>
    // (The JavaScript Text goes here)
</script>
```

- This format allows for other scripting languages as well
- It used to be customary to enclose scripts in HTML comment tags

```
<!-- … -->
```

  - This allowed the script to be ignored by browsers that did not support JavaScript
  - But there is pretty much 100% support for JavaScript now.

Using JavaScript

- You can also pull in the JavaScript source from a URL:

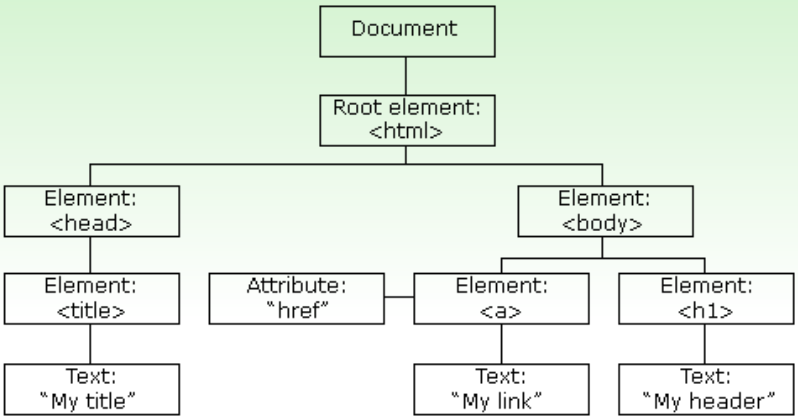
```
<script type="text/JavaScript" src="jquery.js"></script>
```

- In the above case the browser would download the source from a file called jquery.js that would be in the same directory as the html file requesting the JavaScript file.
- For larger projects, this would be the normal approach to take since the JavaScript code can be reused across many web pages on the same site.

What can JavaScript do (1)?

- JavaScript knows about the document it is in, and about elements of that document
- It sees the document as an **object**, and the elements of the document
  - images, forms, etc …as objects as well
- For each object, there are a set of methods that can be applied
  - so
    ```
    document.write()
    ```
  - is an application of a method `write()` inside the `document` object
- One can also declare variables in JavaScript
  - these are entirely untyped
    - and so can contain numbers, text, etc.
- The objects are organised hierarchically into a tree:

HTML Document Object Model (DOM) tree



From: http://www.w3schools.com/htmldom/default.asp

What can JavaScript do (2)

- JavaScript can also
  - Respond to events
  - Check form data
- JavaScript contains many of the programming statements usually associated with full-scale programming languages
  - Functions
  - Variables
  - If … else
  - switch
  - Loops (for and while)
  - And some others as well

JavaScript and accessing parts of a page

- We saw the use of `window.document.write(…)`
  - There is only 1 window, and (generally) one document within it
- If we want to change something in a particular anchor or form or image (or whatever) from a function, we need to identify it

```
<img id="catpic" src="cat.png" alt="A Cat"/>
```

- Then we can refer to it in a function:

```
document.getElementById("catpic").src
document.getElementById("catpic").alt
```

- In this way all the elements of the HTML page are identifiable, readable and writable

Accessing elements inside the HTML

- The Document object has four arrays corresponding to tags that are often repeated:
  - Images
  - Anchors
  - Forms
  - Links
- These are arrays of objects: the correct one may be found either by counting (starting at 0) or by using the name of the tag:

```
<body>
  <a id="lnk" /> Hello, I'm anchor text </a>
```

- We can use either `document.getElementById("lnk")` or `document.anchors[0]` to refer to the same anchor
- The anchor text can also be changed:

```
document.anchors[0].innerHTML = "Hello CSCU9A2";
```


So?

- JavaScript allows dynamic altering of any part of the HTML
  - It allows form input and validation of forms
  - and many other modes of altering the web page
- JavaScript can also be used with the HTML5 *canvas* to create interactive websites and animations [This is covered in the labs.]
- Note however:
  - There can be differences between browser capabilities
  - This can make it very difficult to test and debug JavaScript properly
  - In the labs, you can use the Firebug tool within the Firefox browser to help with debugging.

Server Side vs Client side

- Some internet technologies are **client-side**, and some are **server-side**
- **Client-side**
  - Run on the client machine
    - That is, within the browser
  - Often primarily concerned with display and can be browser dependent (might work on IE8, but not 7, or not Safari, but on Firefox, for example)
- **Server-side**
  - Run on the server
    - That is, on the machine with the rest of the page (and possibly with other data - perhaps a database – as well)
  - Often concerned with storing/retrieving data in databases at server
  - More consistent since it's running in a single environment
- Client side:. Java applets, JavaScript, Cookies
- Server side: PHP, Perl, Java Servlets, ASP, etc

Cookies

- What is a **cookie**?
  - A piece of information sent by a server to a browser
    - stored by a client's browser so it can be
    - sent back from the browser to the server (with the HTTP request) at a later date

Why use cookies?

- Without cookies, browsers are **stateless:**
  - that is: each time you access a site, you always receive precisely the same data
  - No information is held at the browser that identifies a particular user and what they are doing at a particular time
- Cookies provide some **state** information
  - there is some information held at the client browser which can be sent to the server which can be used
    - to alter how the data is displayed
    - to alter what is displayed.
    - (or indeed, for anything else that the programmer at the server decides)

What are cookies used for?

- Storing passwords, preferences, etc. for specific sites
- Marketing information
  - building up a profile of the user
    - what adverts they click on, where they go on the web
  - so that appropriate adverts can be sent to them
    - Google and Facebook do this
- Website tracking
  - although one can count how often different pages have been downloaded, cookies allow the server to determine whether (e.g.) 50 people clicked to see some page or one person clicked 50 times

How do cookies work?

- The server asks the client (browser) to store a cookie value
- The request is embedded within the HTTP header of the page sent from the server to the browser.
  - This is not the same as the <head> of the HTML document.
- When a URL is requested from the same folder as the one that set these cookies up, the cookies are sent (in the HTTP header) with the request for the data to the server
- Cookies may be valid for the current session only (by default), or for some expiration date.
- Cookies may be set so that they are sent to all servers in some domain

Cookie Format

- The information in a cookie consists of
  - Name: the name of the cookie
  - Expiry date
  - Path: the subtree of paths for which the cookie will be valid
  - Value: the value given to the cookie
  - Domain: the domains for which the cookie is valid
- When another page is later loaded say from a relevant domain, the cookies are sent back to the server.
- Note that you cannot see whether cookies are being set by examining the HTML
  - So how do they get set? They can be set using JavaScript.

Cookies, Security and Privacy

- Cookies are not directly insecure
  - they do not lead to the server being able to get at local files, etc.
- But they can give the server information about the user
  - particularly if they are kept over a long period of time
- However, they do allow web page content to be customised for what appears to be the user's benefit
  - or more precisely, for what the designer of the server perceives to be the user's benefit (not exactly the same thing…)
- Is this an invasion of privacy?
  - worries some people who feel their internet activities are being tracked
- Does this provide a hole in browser security?
  - Note that some sites refuse to load if cookies are disabled!
- From 2011, UK law requires websites to obtain explicit consent from users if cookies are to be used.

HTML5 Web Storage

- HTML5 contains a feature called web storage that is a more secure and faster alternative to cookies.
  - Larger storage capacity: 2-10Mb, compared to 4kB for cookies
  - Client side only, so data does not have to be sent to server with each HTTP request – faster, more secure
  - Solves the problem of interference between sessions:
  - Multiple browser windows viewing the same website share the same cookies.
    - Can lead to problems, e.g., buying an item twice
  - HTML5 has a separate data store for each window (session storage)
  - There is also local storage which works like cookies and is shared across windows.

Server-Side Technologies

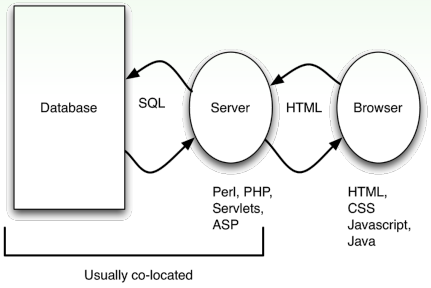
- Aims:
  - Responding to input from the client (browser) apart from simply sending the URL requested
  - Could be
    - Communicating with local resources (e.g. a database)
    - Altering what is sent based on user: specific account details
    - Interpreting cookies
- Technologies:
  - Many different ones
    - Scripting languages, such as Active Server Pages (ASP), Perl, PHP, Python
    - Local programs, e.g. Java Servlets

PHP

- PHP is a server-side cross-platform scripting language
- That is, it runs on the server. It is implemented for a number of server architectures and operating systems
- A method for generating HTML *in response to user actions*
- Used, for example, to generate a new page using the content of forms submitted plus information pulled from a database.

Integrating technologies: e-Commerce example

- E-Commerce uses a database (for prices, stock control etc.), connected to the server. Database access is usually in SQL (Structured Query Language). This is driven by the server, generally as a result of input originated at the browser.
- Different server-side technologies are used: PHP, ASP currently the most common.



End of WWW 2 Lecture

Additional resources:

Learn all about JavaScript at: http://www.w3schools.com/js/

Learn about HTML5 at:
http://www.w3schools.com/html/html5_intro.asp