Michal Bochenek  
EC1401916

**Software Development:   
Developing Websites for Multiplatform Use – Document Object Model   
 Outcome 1**

1. The DOM functionality for a range of multiplatform devices.  
     
   The Document Object Model (DOM) is a way to represent complex documents written in XML, HTML, and XHTML as an aspect of an object model.   
   This model is platform and language independent and it performs based on abstract classes which are then interpreted and organised as a tree structure to provide full control over the elements of this structure (nodes).   
   It is defined by the standards written by the W3C and it defines a group of classes and interfaces allowing to access to the structure of the documents and their modification by creating, removing and modifying nodes.
2. The differences between client and server-side scripting highlighting a couple of examples of the usage of each.  
     
   The client-side environment, which is usually a web browser, is used to run scripts on the website. This takes most of the computing burden away from the server and it is transferred to the end-user’s device in form of a source code being interpreted by the browser.  
     
   Server-side environment that interprets and executes scripting language is a web server, it works on a basis that user sends request which is then granted by generating dynamic HTML page which is then being sent to the client’s web browser. This scripting method is used usually to get an access to the server database and provide interactivity with it, the best example is a web store.  
     
   **Examples of the client-side scripting uses:**  
     
   - JavaScript, Ajax  
   - HTML   
   - CSS   
   - jQuery  
     
   They enhance functionality and user experience by providing graphical effects such as animations, mouse over image effects, form field validation (before submitting it to the server)   
     
   Client-side scripts don’t require additional software to be installed on the server as their interpretation depends only on the web browser capabilities and this is a good choice for web administrators who lack full access to the server on which their web page is stored at.  
     
   There are some security risks involved in client-side scripts, as the access to the system data can be accessed by techniques such as ActiveX but it is not inherently unsafe to run those scripts if the web browser and system are being kept up-to-date.  
     
   Currently there is much more client-side scripting compared with the past, which allows users to experience less refreshing of the web page while he is using.  
     
      
     
   **Examples of the server-side scripting uses:**- all high-level languages such as ASP.net, C#, VB.net, Java, PHP,   
     
   Server-side scripting is used to provide an interface crafted for the specific user, this means that it can use client characteristics to customise the response.  
     
   Every time user generates a query to the server to access new information he has to wait for the response and this can slow down the experience for the user and prevents him to access the webpage while he temporally loses the internet access.  
     
   They run before HTML is loaded, and are executed by the web server when user requests a document as a product of it they produce output in a format understandable by web browsers and send it to the end-user device.  
     
   Server-side scripts require their interpreter to be installed on the web server, and they produce the same output regardless to the client browser, device or system.
3. Suitable methods used to embed scripts within web pages that will work on a range of devices.  
     
   The most common way to embed script within a HTML is to put it directly onto web page code within <script> element this is called inline, but this can become very messy way once web page will grow in size (and code lines) to deal with that a developer can specify an external file source that contains the script code, this will call for this file while loading HTML document in the web browser an example is such as: <script src="myScript.js"></script>.
4. Some of the different scripting languages used and highlight their advantages and disadvantages dependant on the type of device and industry structure.  
     
   Application Service Provider - ASP - is a specification for created website with .asp extension which uses ActiveX scripts, usually Visual Basic Script or Jscript code.   
   This enables programmers to use Visual Basic to develop website and work with tools designed for this language.  
   ASP enables companies to outsource some aspects of their information technology requirements by distributing services to customers from main data centre over internet.  
     
   PHP: Hypertext Pre-processor (PHP3) – it is an open source, server side, HTML embedded language used in creation of the dynamic pages.  
   This language can be embedded within tags, so that author of the page can jump between HTML and PHP, it can perform all tasks which other scripting languages can do but its strongest side is platform compatibility with many types of databases, it can communicate with networks using IMAP, NNTP, SNMP, POP3, HTTP.  
     
   Pearl – Practical Extraction and Report Language – this is a programming language which aims text processing, it is one of the most popular langugaes for writing CGI scripts (Common Gateway Interface).  
     
   Ruby on Rail – its an open source framework for fast creating web apps, it is easy to write in this language as it follows the rule of not repeating yourself and convention over configuration, it also accepts many plug-ins to extend its functions like login, image rescaling or tagging.
5. Code to handle specified events, eg touch screen, push, slide, zoom and key interaction.  
     
   There are two ways of event propagation (order of executing them) in DOM – bubbling and capturing.  
     
   Bubbling is choosing the inner most element event as a first one and then go towards the outermost.  
     
   In capturing there is opposite situation, we start from the outermost and go to the inner element.

To handle specific events we have to create an event handler (listener) and add it to the element such as onclick, onwaiting, onerror etc.  
  
We can then specify method of execution of the even by defining useCapture or useBubbling:   
  
example as a script line:  
  
button.addEventListener("onclick", myFunction, useCapture);  
  
or in line with element:  
  
<button onclick="myFunction()">Click me</button>  
  
This will call for myFunction script once the element was clicked on and it will generate an event for example sound, message notification, generate feedback text etc. event listener feature can be also removed from the element.

1. Current security issues.  
     
   The W3C DOM is following the same security restrictions as DHTML Object Model and is subject to the security rules of Internet Explorer.  
     
   Currently the biggest security issues are coming from the client-side scripting which is enabling to overwrite parts of the script code stored in the end users memory and send it back to the server to breach security and gain unauthorized access.

One of the good examples found is DOM-based cross-site scripting (XSS) attack which relies on improper operating the script in the HTML webpage of the data stored within DOM structure. Once someone gets an access to the object in DOM he might create then a XSS condition by attaching his own script to the structure of the page which will execute on loading.

To prevent DOM-based attack while designing web page we should:  
- avoid client-side actions like rewriting or redirecting using client-side data  
- use of firewalls or scripting which are able to check the inbound URL parameters  
- sanitization of client-side code, especially the one which could be modified