Week2

HTML and the WEB

* HTML is generally processed by web browsers
* The web was designed for ease of publication – non-programmer should be able to develop and deploy web-sites
* To achieve this browsers have traditionally been very accommodating in what they will accept
* Many sites are very long lived - don’t want to break sites just because a new version of HTML, CSS, JS, &c. is available
* So you will see code from lots of versions of the Web & should be in a position to handle it (HTML in general) but we should aim to develop using the latest tools (e.g. HTML5) - particularly because HTML 5 (semantic HTML) added organisational elements that are distinct from previous versions.

HTML

* HyperTextMarkupLanguage
* A language for turning text into hypertext by using markup
  + NB. This is just one way of creating hypertext (it is just perhaps the most dominant)
* The standard markup language for creating web pages
* Not a programming language - no support for programming constructs
* Part of the triad of foundational web technologies (alongside CSS & Javascript)
  + Describes the semantic structure of the data, which CSS presents, and Javascript manipulates
* Browser receives HTML document from server or storage
* Document is then rendered visually (NB. Other user agents may use the returned HTML in other ways)

HTML look like ?

* Plain text - can write HTML in any text editor and only need to save it as a .html document (which can then be opened in a browser)
* Like this

<!DOCTYPE html>  
<html>  
 <head>  
 <title>My first HTML 5 document</title>  
 </head>  
 <body>  
 <p>Hello World from HTML 5</p>  
 </body>  
</html>

W3C

* World Wide Web Consortium (W3C) defines the standards for the web, e.g.
  + HTML 4.01
    - Based on Standard Generalised Markup Language (SGML)
  + HTML 5
    - Backwards compatible but no longer based on (SGML)
* & various related standards: XHTML 1.0, 1.0, 2 (cancelled)
  + Based on eXtensible Markup Language (XML)
* You might see any or all of these versions and related languages whilst working with hypertext systems - we don’t always get to build new stuff from scratch in the latest versions
* NB. The W3C also does a lot more than just define core web standards

Classical to modern

* Until 40.01 defined the visual of a web page
  + Mixed structure and presentation
  + E.g. font face and font size,color size, size of elements, etc
* Modern HTML
  + Describes the content, its structure and its relation to other content
* Visual presentation of those things is delegated to CSS

HTML Elements

* HTML documents are constructed from HTML Elements
* Elements are keywords encapsulated within angle brackets, e.g. <html>
* Elements are represented using opening and closing tags. <html></html>
  + Most tags delineate the start & end of a portion of text or enclose other sets of tags so often paired - one for each end
  + Some stand alone amongst the text e.g <br/>
  + All use angle brackets
* Combined to create a structured documents by denoting structural semantics for the text
* (such as headings, paragraphs, lists, links, etc.)

HTML element structure

Timeline

Description automatically generated

HTML versions

* As you investigate various web pages you will notice considerable variation amongst versions of HTML
* We should write HTML to the current version, e.g. HTML 5, but we should be aware of what earlier versions looked like
* Let’s compare HTML documents from two versions...
  + HTML 4.01 Text, letter

    Description automatically generated
  + HTML 5

Text, letter

Description automatically generated

Validity

* HTML is a language
  + It has a syntactic structure that can be verified as correct (or otherwise)
  + It also has semantics (meaning)
  + There are tools to automatically verify that a given HTML document is correct (or otherwise): <http://validator.w3.org/>

Tags

* Document structure: <html>,<head>,<body>
* Within <head> section: <title>,<base>,<meta>,<style>,<link>
* For text blocks: <address>,<blockquote>,<div>,<H1>…<H6>,<p>,<pre>,<xmp>
* define lists: <dir>, <dl>, <dt>, <dd>,<menu>, <ol>, <ul>, <li>
* define text format: <b>, <basefont>, <big>, <cite>, <code>, <em>, <font>, <i>, <kbd>, <strike>, <sup>, <tt>, <u>, <var>
* define anchors and links: <a>
* define images and image maps: <img>,<area>,<map>
* define tables: <table>, <caption>, <thead>, <tbody>, <tfoot>, <tr>, <th>, <td>
* define forms: : <form>, <fieldset>, <input>, <select>, <option>, <textarea>, <label>, <legend>, <isindex>
* define frames: <frame>, <frameset>,<iframe>
* define scripts: <script>, <noscript>
* define applets & plug-ins:<applet>, <param>, <object> (<embed> not standard)
* Tags that adjust text: <br>, <center>, <hr>

Text formatting

* Headings: <H1>…<H6>
* Physical styles: <b>, <i>
* Logical styles <cite>, <code>, <em>, <strong>
* <font face=”” size “”>
  + NEVER DO THIS
  + Always use CSS for presentation aspects of typography
  + Support for presentational aspects within HTML is slowly being phased out

Lists

* Definition Lists
  + <d1>, <dt>, <dd>
* Ordered lists
  + <ol>, <li>
* Unordered lists
  + <ul>, <li>

Links

* From the Hypertext perspective links are the most important element of HTML
* Hyperlinks turn text into hypertext using two types of link (internal & external):
  + Internal Links
    - Link: <a href=”#name”>…</a>
    - Target: <a name=”name”>… </a>
  + External links
    - To another document in same site: <a href=“page.html”> </a>
    - To a target within another document: <a href=“page.html#name”> ... </a>
    - To another site: <a href=“http://www.simonwells.org”> ... </a>

Tables

* For data representation (not presentation & layout)
  + Although admittedly there is some overlap
* Uses mix of <table>, <tr>, <th>, <td> tags
* Also: <thead>, <tbody>, <tfoot>, <caption> - more semantic structure

Text, letter

Description automatically generated

Images

* <img> with mandatory attributes: src, alt
* Optional attributes::
  + width, height, longdesc
* Image types: GIF, JPG, PNG - browser support is so good that we don’t consider this so much anymore
* NB. Image size

Forms

* Everything so far has been about retrieving HTML pages from the server- using the HTTP GET method
* Sometimes we want to send data from the client to the server (using the HTTP POST method)
  + More commonly using a **for**

Form Controls

* Buttons
  + <input type=”submit”>
  + <input type=”reset”>
  + <input type=”button”>
  + <input type=”image”>
* Check boxes
  + <input type=”checkbox”>
* Radio buttons
  + <input type=”radio”>
* Text boxes
  + <input type=”text”>
* Password textboxes
  + <input type=”password”>
* Hidden fields
  + <input type=”hidden”>
* File upload
  + <input type=”file”>
* Selection lists
  + <select>,<option>,<optgroup>
* Text area
  + <textarea>
* Label(for control)
  + <label>
* Group of controls
  + <fieldset>,<legend>