



# HTML Basics:

## Getting your code to work

Whether you are developing your own profile or program-specific web pages, uploading web-based content onto MyUni for student access or maintaining pages for your School or Faculty web site, a basic understanding of how HTML code works will be invaluable. It will help you identify what might be the cause of problems in the display of page content and enable you to control basic layout and styling features well. These notes are designed to help provide tips for preparing well structured and good looking web content.

### Understanding HTML elements

HTML is a language for 'marking up' ordinary pages of text that can then be opened on your computer or transferred across the Internet to another computer using http (the hypertext transfer protocol for data transfer). When viewed by software that interprets the markup of these 'pages' (a browser or user agent) the contents of the page and any links to resources are displayed. The current version of the language is HTML 4.01 (with Transitional, Strict, and Frames variants). XHTML 1.0 is a version which can be 'read' by both XML and HTML user agents. HTML5 is the latest version (beta) under development and combines features of both HTML 4.01 and XHTML 1.0, but it is not yet widely supported by all browsers. These notes draw on information from all three specifications.

#### *What is an HTML page?*

- HTML files are written in plain text (.txt) but use the extension .html (or .htm)
- The file needs to begin with a Doctype statement which identifies the version of markup language used.
- The rest of the file is wrapped in a root-level element <html>, which contains information about the file <head> and the contents of the web page itself <body>.
- When writing an HTML file 'white space' can be used to space items out for ease of reading, but when the file is interpreted any multiple spaces are reduced to single spaces.
- Standards compliance is the ultimate goal - to mark up your pages so they display consistently in all conforming user agents, and including technologies used by people with disabilities e.g. screen readers.

#### *What are HTML elements?*

- HTML elements are the building blocks of html. They consist of <opening tag> content </closing tag>. <empty tags> don't need closing tags because they don't include 'content' on the page (hr, br, img) Empty tags must close in XHTML to be compliant though, e.g. <br /> Tags need to be 'nested' correctly - they need to close in the reverse order to which they open, e.g. <p><em>emphasised text</em></p> but **not** <p><em>emphasised text</p></em>
- HTML elements can be block elements, where they stack on or inside each other, or inline elements, where they stack alongside each other in the same enclosing block.
- The behaviour of these block and inline elements can be modified using 'style' features such as display, float, z-index, and clip.

## Attributes and values

- Attributes (attribute name="value") provide additional information about an element.
- Standard attributes - core (class, id, style, title), language, and keyboard (accesskey, tabindex)
- Required and optional attributes - e.g. for the <img> tag: alt & src are required, height & width are optional.
- Event attributes - these are to trigger actions (initiate scripts to run) e.g. 'onclick', 'mouseover'

## Styling content

- Styles (and hence stylesheets) are used to modify the visual presentation of HTML elements. The current version of cascading stylesheet specifications is CSS2, but CSS3 is in development.
- Styles will be applied first using any inline styles (at the element level), then internal styles (page level), then as specified in any external stylesheet/s (link, @import), and lastly using the browser's default interpretation.
- If it cannot find a style defined it will cascade to the next source, hence the term cascading stylesheet (CSS).
- The format for styles is: a selector & declaration/s {property: value;}, e.g. li {font-size: 12px; font-color: red;}
- Styles can often be 'inherited' by the 'child' element of a 'parent' element - this is something you will find out more about when you begin to use styles, as it can affect why styles sometimes don't work as expected.

## Web page structure and layout

When you begin marking up your page you first need to consider the structure of the document and semantics (meaning) of the information, and then later consider its visual presentation. There are a number of tools for creating/editing HTML files, from simple programs like *Notepad* (or *Textedit* for MAC), MS Word, web-based editors (as found in MyUni or the in-house TMS), up to WYSIWYG editors like *Adobe Dreamweaver*.

### Page/document structure

- HTML pages contain both unseen <head> (metadata, info about the file or for the file) and visible <body> (the actual page content) data.
- Pages will generally have similar semantic features - header (& hgroup), footer, nav (primary navigation), content (div, article, section), and aside.

### Content elements/types

- The main block level elements include: headings, paragraph, blockquote, lists, tables, forms, address & hr.
- For detailed information on controlling the layout of these elements look up 'CSS visual formatting model'.

### Making connections - hypertext links

- Anchor tags <a href=" " "></a> are used to link to other information or resources internally via 'name' or 'id' attributes and externally to a specified URL. Use the 'title' attribute to provide more information about the link.
- A link can also initiate an email by using the 'mailto:' format, e.g. <a href="mailto:person.name@domain">.
- Styles can be used to modify the appearance of links but their order in the stylesheet is important for their correct behaviour: link, visited, hover, active (remember *Love Hate*, LVHA).

## Modifying text and appearance

Once the overall structure of your document is in place you can then look at how you can enhance it - semantically and visually. Stylesheets and inline styles can be used to modify the appearance of individual or groups of elements.

### Semantic html tags for text

- The most common inline elements include: b, strong, i, em, small, sup, sub, pre, q, cite, abbr, br.

### Special entities and symbols

- For reserved and special characters you need to use the 'escape' character (&) and end with a semicolon (;). e.g. non-breaking space (&nbsp;); © (&copy; or &#169;); small e, acute accent (&eacute; or &#233;); sum  $\sum$  (&sum; or &#8721;); greater or equal  $\geq$  (&ge; or &#8805;)

### Changing text with styles

- CSS properties for modifying elements include such things as: font, color, border, background, text-decoration, list-style, text-align, padding.
- font-size - x-large, pt, px, em, % font-family - name, "new name", generic

- You can use the `<span></span>` tag to apply block-type styles while having it still display inline.
- By using CSS selectors you can change multiple instances over a document, e.g. tag, class, id, name, div/id.
- A key advantage of using stylesheets, apart from making changes to styling easier, is the ability to maintain consistency of appearance across a whole web site.

### *Understanding the box model*

- The box model describes the following properties for an enclosing 'box' on which styles can then be applied: box (content, e.g. text - various properties apply), padding, border, margin. When setting styles for these elements they are applied in the order - top, right, bottom, left.
- The layout behaviour of block elements can be modified using CSS properties such as: float, clear, display, position and z-index.

## **Combining images and multimedia**

Objects such as images, audio and video can be used to include additional information in an HTML page.

### *Including images*

- Suitable image formats for use on web pages include: bmp, gif, png (flat colour), jpg/jpeg (photographic)
- `` (note: xhtml format)
- `<a href="link to other file"><img ... border="0" ></a>`
- Image maps, that define clickable areas on images, can be created with the `<map>` and `<area>` tags.

### *Embedding multimedia*

- The `<object>` tags can be used to embed multimedia and other files such as audio, video, applets, and html.
- The `<param>` tag provides information regarding the initialization of the object - sets values for run-time.
- HTML5 has introduced specific `<audio>` and `<video>` tags.
- For information regarding appropriate formats for multimedia and file size restrictions, for use on University websites, please consult with ITS (Myuni/MyMedia) or the M&C Web team (websites).

## **Styling for accessibility**

Writing your web pages to conform to HTML standards goes a long way to providing accessible content for all people. The Web Content Accessibility Guidelines (WCAG 2.0) help web content developers understand accessibility needs and how to incorporate them into their pages. The guidelines identify three standards of compliance and the University, in line with the Australian Government, requires that our websites comply with the 'AA' standard.

See the University Web guide <http://www.adelaide.edu.au/webguide/content/accessibility/> for more information.

### *Navigation*

- Add 'Skip to content' links - this is included in Uni web templates by default.
- You can improve usability by making all navigation and commands keyboard accessible.

### *Document content*

- Providing 'alt' text for images, audio, video etc. means the user has a fallback description if they fail to load.
- Try to use meaningful links, not just 'click here'. Link titles should tell the user which file they are going to.
- Be conscious of your colour choices and the effect they will have for vision impaired users. There are colour scheme checkers available online that show the level of contrast for common forms of colour-blindness.
- Note that in most browsers you can use the **Ctrl & +** or **-** keys to increase/decrease the size of elements on the page in steps, and the **Ctrl & 0** (zero) keys to restore the default size setting.

### *Linked documents*

- You may need to provide alternate versions of some file types, e.g. PDF, to ensure the content is accessible.

### *Mobile technologies*

- Special consideration also needs to be given to the structure and styling of web pages that will be viewed on mobile technologies such as smart phones, the iPad and other portable devices.

## Restoring order to broken files

Sometimes when we are busily marking up our web page, something goes awry and our 'code' breaks. Some of the common causes to look for and some strategies to employ are detailed below.

### *Working view*

- We like to work in WYSIWYG mode, without all the clutter of the markup tags and attributes, but a good understanding of what you are looking at in code view can be invaluable in helping to solve problems.

### *HTML markup*

- missing end tags, unnecessary end tags, poorly nested tags
- missing or misplaced quotes on attribute values
- links: make sure you have used the exact path or file name, and remember they are case-sensitive.
- <p> tags around content in table cells will add extra spacing within the cell

### *CSS & styles*

- missing semicolon to end declaration
- inheritance - specifying the correct element to modify

### *Validating your code*

- W3C Markup Validation Service - <http://validator.w3.org/>
- W3C CSS Validation Service - <http://jigsaw.w3.org/css-validator/>

## Summary

Well structured and good looking web content doesn't have to be complicated. A basic and informed understanding of what HTML markup is and how it works will make it easier for you to prepare effective and accessible web content, while also giving you the skills to successfully troubleshoot problematic code in your pages.

## References

13. *Objects, Images, and Applets*, (HTML 4.01 Specification. W3C Recommendation 24 December 1999) viewed 5 March 2014, <<http://www.w3.org/TR/1999/REC-html401-19991224/struct/objects.html>>
- Cheat Sheets* (AddedBytes.com), viewed 5 March 2014, <<http://www.addedbytes.com/cheat-sheets/>>
- HTML element*, Wikipedia, viewed 5 March 2014, <[http://en.wikipedia.org/wiki/HTML\\_element](http://en.wikipedia.org/wiki/HTML_element)>
- Ian S. Graham, *The XHTML Language and Design Sourcebook, Extras and Supplementary Material* (7 November 2000) viewed 5 March 2014, <<http://www.iangraham.org/books/xhtml1/extras/>>
- Michael Tuck, *Web Design Principles: HTML: Definition and Characteristics*, viewed 5 March 2014, <<http://www.iraqtimeline.com/maxdesign/basicdesign/principles/prinhtml.html>>
- Ross Shannon, *HTML Source: HTML Tutorials*, viewed 5 March 2014, <<http://www.yourhtmlsource.com/>>
- Steven Estrella, *Appendix B: CSS Quick Reference*, viewed 5 March 2014, <<http://akenbros.com/files/CSS/WebWizardCSSQuickRef.pdf>>
- Vision Australia, *Ten tips to create an accessible website* [web page], viewed 5 March 2014, <<http://www.visionaustralia.org/living-with-low-vision/learning-to-live-independently/using-technology-and-computers/blog---accessibility-and-assistive-technology-blog/blog/accessibility-blog/2012/06/13/ten-tips-to-create-an-accessible-website>>
- Vision Australia, *Digital Access* [web page], viewed 5 March 2014, <<http://www.visionaustralia.org/business-and-professionals/digital-access>>
- W3Schools.com *HTML4 and HTML5 Tutorial*, viewed 5 March 2014, <<http://www.w3schools.com/html/default.asp>>
- W3Schools.com *HTML Entities*, viewed 5 March 2014, <[http://www.w3schools.com/html/html\\_entities.asp](http://www.w3schools.com/html/html_entities.asp)>
- WebAIM, *Web Accessibility for Designers*, viewed 5 March 2014, <<http://webaim.org/resources/designers/>>
- Web Content Accessibility Guidelines (WCAG) 2.0*, W3C Recommendation 11 December 2008, viewed 5 March 2014, <<http://www.w3.org/TR/WCAG20/>>
- Web Content Accessibility Guidelines (WCAG) Overview*, W3C Web Accessibility Initiative, viewed 5 March 2014, <<http://www.w3.org/WAI/intro/wcag.php>>
- Web Design Group, *Structure of an HTML 4 Document*, viewed 5 March 2014, <<http://htmlhelp.com/reference/html40/structure.html>>