Comp 336/436 - Markup Languages

Fall Semester 2017 - Week 3

Dr Nick Hayward

HTML - within the <body> - text formatting

- formatting can be considered relative to stylistic and semantic requirements
- formatting is also available for embedded *code* viewing
- text formatting includes
  - *bold* `<b>`
  - *emphasis* `<em>`
  - *italic* `<i>`
  - *strong* `<strong>`
  - *sub* `<sub>` & *superscripted* `<sup>`
  - *inserted* `<ins>` & *deleted* `<del>`
- computer *code* formatting includes,
  - *code* `<code>`
  - *variables* `<var>`
  - *pre-formatted text* `<pre>`
- quotations, citations and definitions include,
  - *abbreviations* `<abbr>`
  - *acronyms* `<acronym>`
  - *citation* `<cite>`
  - *definition* `<dfn>`

HTML - within the <body> - lists

- list options in HTML
  - *unordered list* `<ul>`
  - *ordered list* `<ol>`
  - *definition list* `<dl>`
- list items `<li>` for `<ul>` and `<ol>`

```
<ul>
  <li>...</li>
</ul>

<ol>
  <li>...</li>
</ol>
```

- definition list uses
  - *`<dt>` for the item*
  - *`<dd>` for the definition*

```
<dl>
  <dt>Super Mario Bros.</dt>
  <dd>iconic platformer...</dd>
</dl>
```

HTML - within the <body> - tables

- organise data within a table
  - *<table> element*
- three primary child elements include
  - *<tr>, <th>, <td>*

```
<table>
  <tr>
    <th>header 1</th>
  </tr>
  <tr>
    <td>row 1, cell 1</td>
  </tr>
</table>
```

- add a <caption>
- span multiple columns using the colspan attribute
- span multiple rows using the rowspan attribute

HTML - metadata & <head> element

- add our CSS styling as either <link> or <style>
- add JavaScript using <script> element

```
<script type="text/javascript" src="assets/default/script.js" />
```

- add <title> of our page
  - *shown in the browser tab or window heading*

```
<title>Our Page Title</title>
```

- <base /> can be used to specify default address or target for all page links

```
<head>
  <base href="http://www.w3schools.com/images/" target="_blank">
</head>
```

- <meta /> adds metadata about the HTML document

```
<meta name="description" content="The Glass Bead Game" />
<meta name="keywords" content="novel, fiction, herman hesse, electronic edition" />
```

HTML - better markup

- web standards are crucial for understanding markup
  - *markup that goes beyond mere presentation*
- improved usage and structure, accessibility, integration...
- with standards, maintenance and extensibility becomes easier
- improved page structure and styling
  - *helps web designers and developers update and augment our code*
- poor markup usage
  - *to achieve a consideration and rendering of pure design*
  - *e.g. nesting tables many levels deep*
  - *adding images and padding blocks for positioning...*
- support for web standards continues to grow in popular browsers
- gives developers option to combine markup and styling
  - *HTML with CSS to achieve greater standards-compliant design*

HTML - markup and standards

- many benefits of understanding and using web standards, e.g.
- *reduced markup*
  - *less code, faster page loading*
  - *less code, greater server capacity, less bandwidth requirements...*
- *separation of concerns*
  - *content, structure, and presentation separated as needed*
  - *CSS used to manage site's design and rendering*
  - *quick and easy to update efficiently*
- *accessibility improvements*
  - *web standards increase no. of supported browsers & technologies...*
- *ongoing compatibility*
  - *web standards help improve chances of compatibility in the future...*

HTML - better structure

- consider *semantic* or *structured* markup
  - *within the context of app usage and domain requirements*
- trying to impart a sense of underlying meaning with markup
  - *correct elements for document markup*
- for a list
  - *use correct list group with list items - e.g.* `ul`, `li`...
- for a table
  - *consider table for data purposes*
  - *structure table & then consider presentation...*
- *semantic* markup helps create *separation of concerns*
  - *separate content and presentation*
  - *improves comprehension and usage*

HTML - XHTML

> *"The XHTML family is the next step in the evolution of the Internet. By migrating to XHTML today, content developers can enter the XML world with all of its attendant benefits, while still remaining confident in their content's backward and future compatibility." (W3C - www.w3.org/TR/xhtml1/#xhtml)*

- greater chance of comprehension, correct rendering...
- XHTML rules and valid markup

HTML - XHTML validation

- XHTML introduces many rules from XML, e.g.
  - *tags and attributes must be in lowercase*
  - *attributes must be enclosed with quotes*
  - *all tags must eventually close*
  - *no overlapping hierarchies*
  - *...*
- XHTML benefits include,
  - *improved standards compliancy for sites and apps*
  - *greater ease of sharing and collaboration*
  - *easier code debugging*
  - *opportunities for markup validation and compliance*
  - *improved search, archiving, rankings...*
- W3C provides an online validation tool
  - *http://validator.w3.org/*

Semantic HTML - intro

- importance of web standards
  - *and their application to HTML markup and documents*
- standards help drive a consideration of markup, e.g. HTML
  - *usage for what they mean*
  - *not simply how they will look...*
- semantic instead of purely presentational perspective
  - *introduction of meaning and value to the document*
- when pages are processed
  - *impart structure and meaning beyond mere presentation*
- a core consideration for usage of markup languages
- issues persist with HTML element usage
  - *e.g. inline elements such as `<b>` and `<i>`*

Semantic HTML - a reason to care

- Semantic HTML - opportunity to convey meaning with your markup
  - *meaning may be explicit due to the containing element*
  - *implicit due to a structured grouping of elements*
- markup makes it explicit to the browser
  - *underlying meaning of a page and its content*
- notion of meaning and clarity also conveyed to search engines
  - *fidelity with query and result...*
- semantic elements provide information beyond page rendering and design
- use semantic markup correctly
  - *create more specific references for styling*
  - *greater chance of rendering information correctly*

Semantic HTML - example usage

```
<!-- incorrect element chosen -->
<div id="code">
document.addEventListener('click', function () {
  console.log('Click received...');
});
</div>
```

```
<!-- correct element chosen -->
<code>
document.addEventListener('click', function () {
  console.log('Click received...');
});
</code>
```

- semantic example usage

Semantic HTML - correct usage

- need to ensure elements convey their correct meaning
  - *i.e. the meaning expected for the contained content*
- e.g. often see the following elements mis-used and applied incorrectly for markup,
  - *<p> - paragraphs*
  - *<ul - unordered list*
  - *<h1> to <h6> - headings*
  - *<blockquote> - blockquote*
- using <blockquote> to simply help indent text
  - *instead of CSS margins...*
- or the perennial mis-use of a <p>
  - *simply add extra space between elements*

```
<p> <p>
```

HTML - structure & validation - example

Using lists correctly...

```
<li>nice</li>
<li>cannes</li>
<li>menton</li>
```

- list markup looks OK
  - *still fails validation for an obvious reason*
  - *missing structural grouping for list items*
  - *not valid markup...*
- semantics of the overall list are missing
- example - basic list items

HTML - a semantic point of view

```
<ul>
  <li>nice</li>
  <li>cannes</li>
  <li>menton</li>
</ul>
```

- from the perspective of semantics
  - *meant to act as a group of items that belong together*
- denote such groupings with correct semantic markup
- structuring items to clearly denote their meaning and purpose
- consider global attributes
  - *https://developer.mozilla.org/en-US/docs/Web/HTML/Global_attributes*
- example - basic group

HTML - semantics & XML

```
<places>
 <item>nice</item>
 <item>cannes</item>
 <item>menton</item>
</places>
```

- XML example of markup for places group and items
- entire list has a containing element, `<places>`
- grouping items has a number of benefits
  - *comprehension, legibility*
  - *ease of processing...*
- XSLT processing into HTML, plain text, PDF...
- XML principles incorporated into XHTML and HTML5
- example - basic xml list

HTML - benefits of structure & validation

- define and create a meaningful structure for required markup
  - *improves usage and flexibility as project develops*
  - *provides extensible structure for project*
- for example, benefits include
  - *helps increase ease of CSS styling*
  - *creates properly structured documents*
  - *improves general management of updates to markup*
  - *...*
- easier to understand and easier to maintain and update
- structured, valid markup aids in repurposing data
  - *into various representations of information*

HTML - benefits of structure & validation - example 1

e.g. a standard list

```
<ul>
  <li>nice</li>
  <li>cannes</li>
  <li>menton</li>
  <li>antibes</li>
  <li>grasse</li>
</ul>
```

- example - basic group style

HTML - benefits of structure & validation - example 2

e.g. lists for navigation, menus, tabs...

```
<ul id="menutabs">
  <li><a href="nice">nice</a></li>
  <li><a href="cannes">cannes</a></li>
  <li><a href="menton">menton</a></li>
  <li><a href="antibes">antibes</a></li>
  <li><a href="grasse">grasse</a></li>
</ul>
```

- example - basic menu tabs

HTML - markup for headings - part 1

- HTML is flexible in markup usage
  - *due to presentational versus structural considerations*
- headings might be perceived as purely presentational, e.g.

```
<span class="heading">Chapter 1</span>
```

- issues with presentational markup, e.g.
  - *visual browsers with CSS will render as expected*
  - *no CSS, and browsers will render as normal text*
  - *non-visual browsers = normal text and no heading*
  - *accessibility issues...*
- search engines, ranking, spiders...
  - *will not process this markup as a heading*
  - *no semantic meaning...*
  - *recorded as normal text*
- CSS styles can be unique
  - *but restricted to `class` usage with `heading`*

HTML - markup for headings - part 2

- many different ways to markup content with HTML, e.g.

```
<p><b>Chapter 1</b></p>
```

- issues still exist with variant markup options, e.g.
  - *visual browsers will render text in bold & same size as default*
  - *unique styling is problematic...*
  - *search engines, ranking, spiders...*
    - will not process this markup as a heading
    - no semantic meaning...
    - recorded as normal text

HTML - markup for headings - part 3

- use markup correctly with structure and meaning, e.g.

```
<h3>Chapter 1</h3>
```

- benefits of this markup, e.g.
  - *conveys meaning to contained text*
  - *visual and non-visual browsers treat heading correctly*
    - regardless of any associated styles...
  - *easy to add unique styles with CSS*
  - *search engines &c. will interpret this markup correctly*
    - extract keywords, semantics, structure...

HTML - markup for tables

- great example of poor usage of HTML markup is `<table>` element
- main issue is use of nested tables and spacer elements, images...
- if used correctly in structured markup
  - *tables can be very useful structure*
  - *impart a sense of semantic organisation to data*
  - *creating various interpretive information*
- what is a table for?
  - *structuring data*
  - *data to impart curated information...*

HTML - markup for tables - example I

- simple table example - columns and rows for *presentation* purposes

```
<p>Travel Destinations</p>
<!-- basic table structure - minimal - rows and columns -->
<table>
  <tr>
    <td><b>Place</b></td>
    <td><b>Country</b></td>
    <td><b>Sights</b></td>
  </tr>
  <tr>
    <td>Nice</td>
    <td>France</td>
    <td>Cours Saleya</td>
  </tr>
  <tr>
    <td>Cannes</td>
    <td>France</td>
    <td>La Croisette</td>
  </tr>
  <tr>
    <td>Antibes</td>
    <td>France</td>
    <td>Picasso museum</td>
  </tr>
</table>
```

example

- example - basic table for presentation

HTML - markup for tables - example 2

- add semantic structure & elements to table caption - replace <p> with correct <caption> usage for a table...

```
<!-- basic table structure - minimal - add a caption -->
<table>
  <caption>Travel Destinations</caption>
  ...
```

- modern browsers style <caption> by default
  - *centred above the table*
- modify styling as required

example

- example - basic table caption

HTML - markup for tables - example 3

- add a summary attribute to the table

```
<!-- basic table structure - minimal - add summary attribute -->
<table summary="structured table data for travel destinations...">
  <caption>Travel Destinations</caption>
  ...
```

- add further meaning and structure to the table
  - *use of a summary attribute on the table element*
- processed by the browsers for semantics
- particularly useful for non-visual browsers

example

- example - basic table with summary

HTML - markup for tables - example 4

- add correct headers <th> to the table

```
<!-- basic table structure - minimal - add table headers -->
<table summary="structured table data for travel destinations...">
  <caption>Travel Destinations</caption>
  <tr>
    <th>Place</th>
    <th>Country</th>
    <th>Sights</th>
  </tr>
  ...
```

Benefits include:

- remove need for presentational markup, bold elements
- visual browsers process structural and presentation qualities of headings
- such heading elements can also be useful for non-visual browsers

example

- example - basic table with headers

HTML - markup for tables - example 5

- table markup and accessibility markup...

```
<!-- basic table structure - accessibility - add ids and headers -->
<table summary="structured table data for travel destinations...">
  <caption>Travel Destinations</caption>
  <tr>
    <th id="place">Place</th>
    <th id="country">Country</th>
    <th id="sights">Sights</th>
  </tr>
  <tr>
    <td headers="place">Nice</td>
    <td headers="country">France</td>
    <td headers="sights">Cours Saleya</td>
  </tr>
  ...
```

- creating a known relationship between the table's header, and its data
- a screen reader, for example, may read this table as follows,
  - *Place: Nice, Country: France, Sights: Cours Saleya*
- established a pattern to the output information for non-visual devices...

example

- example - basic table with accessibility

HTML - markup for tables - example 6

- add extra semantic markup for thead, tfoot, tbody...

```
<!-- basic table structure - add head, foot, body -->
<table summary="structured table data for travel destinations...">
  <caption>Travel Destinations</caption>
  <thead>
    <tr>
      ...
    </tr>
  </thead>
  <tfoot>
    <tr>
      ...
    </tr>
  </tfoot>
  <tbody>
    <tr>
      ...
    </tr>
  </tbody>
</table>
```

- head and foot elements customarily go above the table body
  - *allows modern browsers, readers, &c. to load that data first*
  - *then render the main table content*

Benefits include:

- better underlying structure to data
- greater ease for styling a table due to clear divisions in data and information
- structural and presentational markup now working together correctly...

example

- example - basic table with head, foot, body

HTML - presentational vs structural

- consider *presentational* vs *structural*
  - *e.g.usage of quotations in markup*
  - *similar consideration to headings...*
- need to convey meaning and structure
- rather than a mere presentational instruction
- consider HTML's phrase elements
  - *e.g.* `<cite>`, `<code>`, `<abbr>`
- each phrase element imparts a sense of underlying meaning
  - *structure & then presentation...*

HTML - minimising markup

- noticeable benefit to creating sites with valid markup
  - *separation of structural from presentational*
  - *general reduction in required markup*
- simply conforming to the W3C's specifications
  - *does not inherently guarantee less code for your project*
  - *possible to include many unnecessary elements & retain valid markup*
  - *markup may still be valid*
- project issues may include:
  - *lack of efficiency*
  - *extraneous markup and code*
- to help minimise markup
  - *consider classes added to markup*
    - are there too many? are they all necessary? &c.
    - avoid class usage for unique reference
  - *avoid `<div>` usage for explicit block-level elements* ***

XML - intro

- XML = EXtensible Markup Language
- markup language similar to HTML
- designed to carry data but not display data
- XML tags are not pre-defined, you must define your own
- designed to be self-descriptive
- XML is a W3C recommendation

XML - simple example

```
<calendar>
  <date>4th November, 1922</date>
  <title>Tomb of Tutankhamun discovered</title>
  <location>Valley of the kings, Luxor, Egypt</location>
  <kv>62</kv>
</calendar>
```

XML - separation of data

- XML allows a designer to separate aspects of data, e.g.
  - *data that needs updating on a regular basis*
  - *from HTML code necessary to display content*
- XML can be stored in separate files
  - *updated as necessary without affecting the HTML*
- XML also helps simplifies data sharing
  - *stored in a plain text format*
  - *platform agnostic*
- easy to exchange XML files and data

XML - migration of data

- XML eases migration to new development platforms, languages systems...
- data can be available to all kinds of *reading machines*
  - *handheld computers, voice machines, news feeds, &c.*
  - *ease of provision for accessible devices and services*
- XML also used as base for creation of other internet technologies, e.g.
  - *XHTML*
  - *RSS*
  - *RDF*
  - *OWL*

XML - structure

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<calendar>
  <date>4th November, 1922</date>
  <title>Tomb of Tutankhamun discovered</title>
  <location>Valley of the kings, Luxor, Egypt</location>
  <kv>62</kv>
</calendar>
```

- basic structure

XML - structure - syntax

- there must be closing tags, e.g.

```
<p>a new paragraph...</p>
```

and not

```
<p>a new paragraph...
```

- tags are case-sensitive
- elements must be properly nested, e.g.

```
<bold><italic>a new phrase...</italic></bold>
```

and not

```
<bold><italic>a new phrase...</bold></italic>
```

- must have a root element
- attribute values must be quoted
- entity references e.g. using a character such as < instead of `<`

XML - structure - entity references

character	reference	meaning
<	<	less than
>	>	greater than
&	&	ampersand
'	'	apostrophe
"	"	quotation mark

XML - structure - comments

```
<!-- tomb discovered by Howard Carter -->
```

Demos

HTML

- basic list items
- basic group
- basic group style
- basic menu tabs
- basic table presentation
- basic table caption
- basic table with summary
- basic table with headers
- basic table with accessibility
- basic table - head, foot, body
- semantic usage

XML

- basic XML
- basic structure

References

- MDN - HTML Block-level vs Inline
- MDN - HTML <b> element
- MDN - HTML Global Attributes
- MDN - HTML Heading elements
- MDN - HTML <table> element
- MDN - HTML <ul> element
- Wikipedia - list of XML markup languages